Continuous learning augmented investment decisions

Daniel Philps
Department of Computer Science
City, University of London
daniel.philps@city.ac.uk

Tillman Weyde
Department of Computer Science
City, University of London
t.e.weyde@city.ac.uk

Artur d'Avila Garcez
Department of Computer Science
City, University of London
a.garcez@city.ac.uk

Roy Batchelor
Cass Business School
City, University of London
r.a.batchelor@city.ac.uk

Abstract

Investment decisions can benefit from incorporating an accumulated knowledge of the past to drive future decision making. We introduce Continuous Learning Augmentation (CLA) which is based on an explicit memory structure and a feed forward neural network (FFNN) base model and used to drive long term financial investment decisions. We demonstrate that our approach improves accuracy in investment decision making while memory is addressed in an explainable way. Our approach introduces novel remember cues, consisting of empirically learned change points in the absolute error series of the FFNN. Memory recall is also novel, with contextual similarity assessed over time by sampling distances using dynamic time warping (DTW). We demonstrate the benefits of our approach by using it in an expected return forecasting task to drive investment decisions. In an investment simulation in a broad international equity universe between 2003-2017, our approach significantly outperforms FFNN base models. We also illustrate how CLA's memory addressing works in practice, using a worked example to demonstrate the explainability of our approach.

Keywords: Continuous learning, explicit memory, neural network, investment, trading.

1 Introduction

Time-series are ubiquitous in modern human activity, not least in finance, and applying continuous learning to financial time-series problems would be highly beneficial. However, existing approaches suffer from problems such as catastrophic forgetting, the opacity of implicit memory or outright complexity , arguably making them not well suited to the high magnitude decision making that tends to be required in finance. We introduce Continuous Learning Augmentation (CLA), a time-series based memory approach that accumulates models of past states, which are recalled and balanced when states approximately reoccur. We apply CLA to drive investment decisions in international equities markets, the results of which show improved investment performance compared to a feedforward neural network (FFNN) base model. (In further research we have found that CLA outperforms more traditional, linear finance models also). We also show a worked example how CLA's memory addressing adds explainability.

The remainder of this paper is organized as follows. The next section reviews related work on memory modelling. We then introduce CLA in Section three, describing how different states are

identified and remembered, and how models of past states are recalled and applied, as well as how less relevant models are forgotten. In Section four, we report experiments with feedforward neural networks that show significant improvements with CLA. In Section five, we illustrate how CLA enables interpretation in terms or references to past states. Finally, we present our conclusions and directions for future work.

2   Related work

Approaches developed for continuous learning have been applied to many areas and have employed a wide range of techniques including gated neural networks [17][2], explicit memory structures [37], prototypical addressing[33], weight adaptation [16][34] to name a few. Having addressed catastrophic forgetting [11] more recent research has turned to solving second order problems, such as overheads of external memory structures [28], problems with weight saturation [24] and the complications of outright complexity [38]. While a number of memory approaches apply to sequential memory tasks [13][14] a far smaller number still have been focused specifically on time-series [21][12][25][10]. It is unclear how effective these approaches would be in dealing with long term continuous learning of noisy, non-stationary time-series, commonly found in finance.

2.1   Financial regimes as a memory concept

Regime switching models and change point detection provide a simplified answer to identifying changing states in time-series with the major disadvantage that change points between regimes (or states) are notoriously difficult to identify out of sample [8] and existing econometric approaches are limited by long term, parametric assumptions in their attempts [6, 4, 35, 29, 7, 40, 32]. There is also no guarantee that a change point in a time-series represents a significant change in the accuracy of an applied model, a more useful perspective for modelling different states. Another approach is to focus on a model's absolute error, aiming to capture as much information as possible regarding changes in the relation between independent and dependent variables [15]. Different forms of residual change have been developed [1, 20, 19, 26, 18]. However, most approaches assume a single or known number of change points in a series and are less applicable to a priori change points or multivariate series [15].

2.2   Memory models

Memory modelling approaches using external memory structures requires an appropriate memory addressing mechanism (a way of storing and recalling a memory). Memory addressing is generally based on a similarity measure such as cosine similarity [13, 27] kernel weighting [36], use of linear models [33] or instance-based similarities, many using K-nearest neighbours [22] [34]. However, these approaches are not obviously well suited to assessing the similarity of noisy and non-stationary, multivariate time-series. Euclidean distance offers a way to compare time-series but has a high sensitivity to the timing of data-points, something that has been addressed by dynamic time warping (DTW) [30, 5]. However, DTW requires normalized data [23] and is also computationally expensive, although some mitigating measures have been developed (see [39]).

3   Continuous Learning Augmentation

Continuous learning augmentation (CLA) is a regression approach applied as a sliding window stepping forward through time, over input data of one or more time-series. The approach is initialized with a an empty memory structure, M and a FFNN base model, ϕ, parameterized by θ_B. The base model is applied to a multivariate input series, X with K variables over T time steps. The base models produces a forecast value $\hat{y_{t+1}}$ in each period as time steps forward. A *remember* function, j, appends a new model-memory, M_t^m, to M, on a *remember cue* defined by the change in the base model's absolute error at time point t. A *recall* function g balances a mixture of base model and model-memory forecasts.

Figure 1 shows the functional steps of remembering and recalling model-memories.
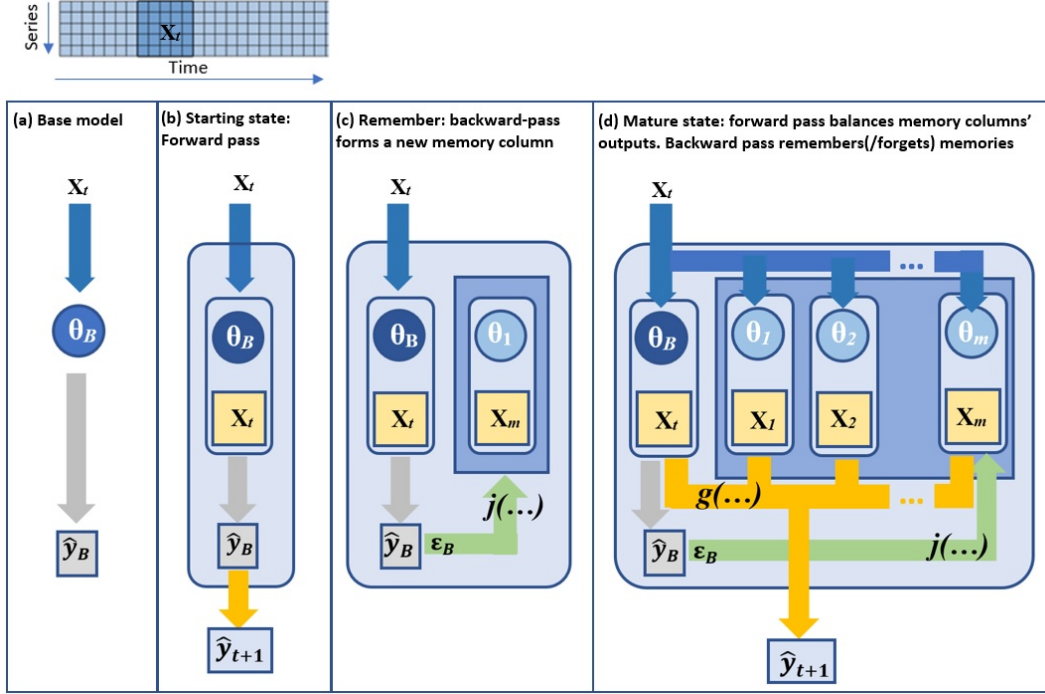
Figure 1: Continuous learning augmentation architecture. (a) A regression base model, ϕ, parameterized by θ_B, is run, stepping forward through time, training at each time step. (b) This base model is structured as a column containing model parameters, θ_B, and a contextual reference, X_t. Initially the base model is run as normal, completing a forward pass with the input data X_t to forecast ($haty_{t+1}$). (c) As time steps on, y becomes observable and a backward pass is conducted where the absolute error of the base model, ϵ_B, determines if a change point has occurred, On a change the function j copies the base model column to a new memory column in M. The base model is then trained. (d) Over time more change points will be detected and more memory columns will be added. At each time step, all memory columns are run using the current input data X_t. The forecast results of all the columns (including the base model column) are balanced by the function g which uses the similarity of the current input, X_{t-1} with the contextual reference of each memory, X_m, to weight the output of each column, $\hat{y_B}$ and $\hat{y_m}$ of M, to result in $\hat{y}_{(t+1)}$).

3.1 Memory management

Repeating patterns are required in the input data to provide memory cues to remember and recall different past states. Model parameters trained in a given past state, θ_m, can then be applied if that state approximately reoccurs in the future. When CLA forms a memory, it is stored as a column in an explicit memory structure, similar to [3], which changes in size over time as new memories are remembered and old ones forgotten. Each memory column consists of a copy of a past base model parameterization, θ_m, and the training data X_m used to learn those parameters; (X_m, θ_m). As the sliding window steps into a new time period, CLA recalls one or more model-memories by comparing the latest input data (X_t) with the training data stored in each memory column (X_m). Memories with training data that are more similar to the current input series will have a higher weight applied to their output ($\hat{y}_{m,t+1}$) and therefore make a greater contribution to the final CLA output ($\hat{y}_{t+1}$).

3.2 Remembering

Remembering is triggered by changes in the absolute error series of the base model, ϵ_B, as the approach steps forward through time. These changes are assumed to be associated with changes in state which are indicated by a function j. j defines a change and stores a pairing of the parameterization

of a base model, θ_B, and a contextual reference, $\boldsymbol{X_t}$. Figure 1c) shows how a change is detected by function j from a backward-pass, which then results in a new memory column being appended to $\boldsymbol{M}$:

$$\boldsymbol{M} = \big\{ (\boldsymbol{X_1}, \theta_1), \dots, (\boldsymbol{X_m}, \theta_m) \big\} \tag{1}$$

Immediately after the remember event has occurred, a new base model is trained on the current input, overwriting θ_B.

Theoretically, for a fair model of a state, ϵ_B would be approximately $i.i.d.$ with a zero valued mean. Therefore the current base mode would cease to be a fair representation of the current state when ϵ_B exceeds a certain confidence interval, in turn implying a change in state. J_{Crit} represents a critical level for ϵ_B, indicating a change point has occurred in state. Memories are only stored when the observed absolute error series,ϵ_B, spikes above a critical level, J_{Crit}:

Algorithm 1 Remember function j

Require: Initialize memory structure $\boldsymbol{M}$
Require: Initialize J_{Crit}
   # Step through time, period by period, starting at the earliest date
   for all time steps t in T **do**
      $\epsilon_{B,t} \leftarrow$ error after forward pass of base model $\phi(\boldsymbol{X_t}, \theta_B)$
      if $|\epsilon_{B,t}| \geq J_{Crit}$ **then**
         append model-memory $(\boldsymbol{X_t}, \theta_B)$ to $\boldsymbol{M}$
      end if
      # Dependent variable becomes observable
      $\theta_B \leftarrow$ train base model and overwrite $(\boldsymbol{X_t}, \theta_B)$
      $J_{Crit} \leftarrow$ learn and update J_{Crit}
   end for

J_{Crit} is a hyperparameter, optimized at every time step, to result in a level of sensitivity to remembering that forms an external memory, $\boldsymbol{M}$, resulting in the lowest empirical forecasting error for the CLA approach over the study term up until time T:

$$J_{Crit} = \underset{j_{Crit} \in jgrid}{\operatorname{argmin}} f(\boldsymbol{X_t}, j_{Crit}) \tag{2}$$

Where f is the CLA approach expressed as a function of the input series and j_{Crit}, yielding $\epsilon_{B,t}$, the absolute error of the base model at time t and $jgrid$ is a 20 point, equidistant set between the minimum and the maximum values of ϵ_B.

3.3 Recall

The recall of memories takes place in the function g, which calculates $\hat{y}_{m,(t+1)}$ a mixture of the predictions from the current base model and base models from model-memories.

$$\hat{y}_{(t+1)} = g(\boldsymbol{X_t}, \boldsymbol{M^t}) \tag{3}$$

The mixture coefficients are based on comparing the similarity of the current time varying context $\boldsymbol{X_t}$ with the contextual references $\boldsymbol{X_m}$ stored with each individual memory. The mixture memories that are more similar to the current context have a greater weighted in CLA's final modelling outcome. Dynamic time warping (DTW) is used to calculate contextual similarity. However, multivariate DTW is computationally expensive [31]. As well as applying traditional constraints to the warping path, we also use a sampling based implementation to reduce expense further. DTW is only applied to a subset of N randomly sampled instances from $\boldsymbol{X_m}$ and $\boldsymbol{X_t}$, sampling over rows, each of which represent different securities in the dataset:

$$\hat{D}(\boldsymbol{X_m}, \boldsymbol{X_t}) = 1/N \sum_{i=0}^{N} DTW(X_{m,r_1(D)}, X_{t,r_2(D)}) \tag{4}$$

4

Long/Short – *Best* – FFNN (2003-2017) simulation tests

			Total Return					Augmentation benefit (v Base model)				
			TR (%)	SD (%)	TR/SD	t-stat	p-val	RR [%]	SD (%)	RR/SD	t-stat	p-val
Min	50	187	-1.84%	3.83% -	0.48 -	1.90	5.90%	-5.98%	2.66% -	2.24 -	8.86	0.00%
Median	50	187	**4.23%**	**4.68%**	0.90	3.57	**0.05%**	1.47%	4.32%	0.34	1.34	18.18%
Mean	50	187	**4.32%**	**5.29%**	0.82	3.22	**0.15%**	1.52%	4.80%	0.32	1.25	21.30%
Max	50	187	11.06%	6.77%	1.63	6.45	0.00%	8.36%	10.30%	0.81	3.20	0.16%

	pctPos	pctNeg	p-val
Augmentation benefit signs	76.5%	23.5%	0.0%

Long/Short – *Similarity Weighted* – FFNN (2003-2017) simulation tests

			Total Return					Augmentation benefit (v Base model)				
	Obs (T		TR (%)	SD (%)	TR/SD	t-stat	p-val	RR (%)	SD (%)	RR/SD	t-stat	p-val
Min	50	187	3.75%	3.60%	1.04	4.11	0.01%	-0.72%	3.63% -	0.20 -	0.78	43.63%
Median	50	187	**7.18%**	**4.20%**	1.71	6.75	**0.00%**	4.24%	4.44%	0.96	3.77	**0.02%**
Mean	50	187	**7.10%**	**4.28%**	1.66	6.55	**0.00%**	4.06%	4.89%	0.83	3.28	**0.12%**
Max	50	187	9.14%	6.25%	1.46	5.77	0.00%	7.33%	11.74%	0.62	2.46	1.46%

	pctPos	pctNeg	p-val
Augmentation benefit signs	94.1%	5.9%	0.0%

Figure 2: Long/short investment simulations. 50 simulations were run using a CLA with g_{best} (*best*) balancing and then using $g_{SimWeight}$ (*similarity weighted*). For each test the min, max, mean and median total return results are shown. TR is the annualized total return, SD the annualized standard deviation, TR/SD is the Sharpe ratio. Augmentation benefit shows the performance of CLA over the base model where RR is the annualized relative return of CLA over the base model and RR/SD is the information ratio. P-values of the t-stats of the Sharpe and information ratios show that CLA produces statistically significant Sharpe at the 1% level, for the mean and median results and statistically significant information ratios, at the 1% level, for the *similarity weighted* simulations. The signs tests show statistically significant hit rates at the 1% level for both *best* and *similarity weighted* tests.

Where $\hat{D}$ is the expected distance, N is the number of samples to take and $r_1(D), r_2(D)$ are random integers between 1 and D.

The mean, sampled distance is used to determine the similarity between the current context and those of each memory.

3.4 Balancing

Two different approaches to memory balancing were used, firstly, the *best individual* (i.e. lowest distance) model-memory:

$$\hat{y}_{t+1} = g_{Best}(\boldsymbol{X_t}, \boldsymbol{M_t^m}) \tag{5}$$

where g_{Best} is an output function to select the model-memory which is most similar to the current context (i.e. $\underset{m}{\operatorname{argmin}}\hat{D}(\boldsymbol{X_m}, \boldsymbol{X_t})$), $\hat{y}_{t+1}$ is the regression output. Secondly, a *similarity weighted* ensemble of all model-memories, $g_{SimWeight}(\boldsymbol{X_t}, \boldsymbol{M})$:

$$\hat{y}_{t+1} = \sum_{m=1}^{M} \phi(\boldsymbol{X_t}, \theta_m) \cdot \left[1 - \frac{\hat{D}(\boldsymbol{X_m}, \boldsymbol{X_t})}{\sum_{m=1}^{M} \hat{D}(\boldsymbol{X_m}, \boldsymbol{X_t})} \right] \tag{6}$$

Where M is the number of memories in the memory structure $\boldsymbol{M}$. As a past state is unlikely to perfectly repeat, defining this continuous function for balancing model-memories is more likely to generalize better [14] than picking the best single model (which is indeed found to be the case).

4 Simulating investment decisions

CLA is used in a regression task to forecast future expected returns of individual equity securities and used to drive an equities investment simulation. Stock level characteristics were used as the input

dataset, to batch train an FFNN over all stocks in each period, forecasting US$ total returns 12months ahead for each stock. Where a forecast was in the top(bottom) decile it was interpreted as a buy(sell) signal.

Stock level characteristics are commonly expressed using factor loadings [9]. These were estimated, in-sample at each time step by regressing style factor excess returns against each stock level US$ excess return stream: $r_{i,t} = \alpha_{i,t} + \beta_{MKT,i,t}x_{MKT,t} + \beta_{VAL,i,t}x_{VAL,i,t} + \epsilon_{i,t}$, where $r_{i,t}$ is the excess return of stock i in period t, $x_{MKT,t}$ is the excess return of the All Countries World Ex-USA Equities Index, $x_{VAL,t}$ is the relative return of the All Countries World Ex-USA Value Equities Index.

Stock level factor loadings populated a matrix X, which comprized the input data. Each row represented a stock appearing in the index at time t (up to 4,500 stocks) and each column related to a coefficient calculated on a specific time lag. X resulted from winsorizing the raw input to eliminate outliers. Training of a single FFNN, latent variable model per period, against all stock level data, used training, cross validation and testing sets in 75/5/25 proportions. Long/short model portfolios were constructed every six months over the study term, simulating a rebalance every 6months, using equal weighted long positions (buys) and shorts (sells). The simulation encompassed 4,500 international equities in total, covering over 30 countries across developed and emerging markets, corresponding to the the All Countries World Ex-USA Equities Index between 2001-2017. (Note that the first 24months were used as a training period while testing, which was entirely out of sample and free from known data snooping biases, started in 2003). To account for the DTW sampling approach used, multiple test runs were carried out. 50 simulations were run per test for this purpose. Both *best* and separately *similarity weighted* were tested. Further testing was conducted to investigate whether results exhibited only an ensemble effect. An equal weighted balancing approach was also tested and found to generate weaker positive total returns relative to both *best* and the *similarity weighted* balancing approaches, demonstrating that CLA is exhibiting more than an ensemble effect

5 Simulation results

5.1 Accuracy: Investment decisions

CLA results for long/short simulations showed a significant return benefit over the FFNN base model. (in further testing it was also found CLA produces positive and statistically significant augmentation benefits to linear models also.) It is also found that CLA is particularly effective at identifying stocks that produce a poor future return. As well as producing good performance relative to the base model, CLA also produced strong positive hit rates, statistically significant to the 5% level or better in both tests. Examining the distribution of simulation results it is notable that as the base model error increases, augmentation benefit also increases, indicating a stronger augmentation benefit when the base model error is higher and vice versa. This property was exhibited in all tests conducted.

5.2 Explainability: Interpretable Memory

CLA produces results that can be explained by examining which past models have been applied to which approximately repeating states, and with further investigation, why. Figure 3 shows an example of a simulation run, where 3a) shows how the value of $1 would have changed if invested in an investment strategy driven by CLA and, separately, by the base model. Figure 3b) shows the memory structure of CLA, where a new memory can theoretically be appended at every step forwards in the simulation, although only four memories were remembered in this example. Two memories are examined. Firstly, a memory formed in the period ending July 2006, which is used by CLA to outperform the base model in the period Sep 2007 - Oct 2008. Interestingly this is over the period of the *Quant Quake* and up until just after the collapse of Lehman Brothers in the thick of the 2008 Financial crisis. Secondly, a memory formed in the period ending December 2009 is used by CLA between 2011 and mid 2014, the period affected by the euro zone crisis and subsequent recovery. It is also used by CLA, albeit to far lesser effect, in late 2016.
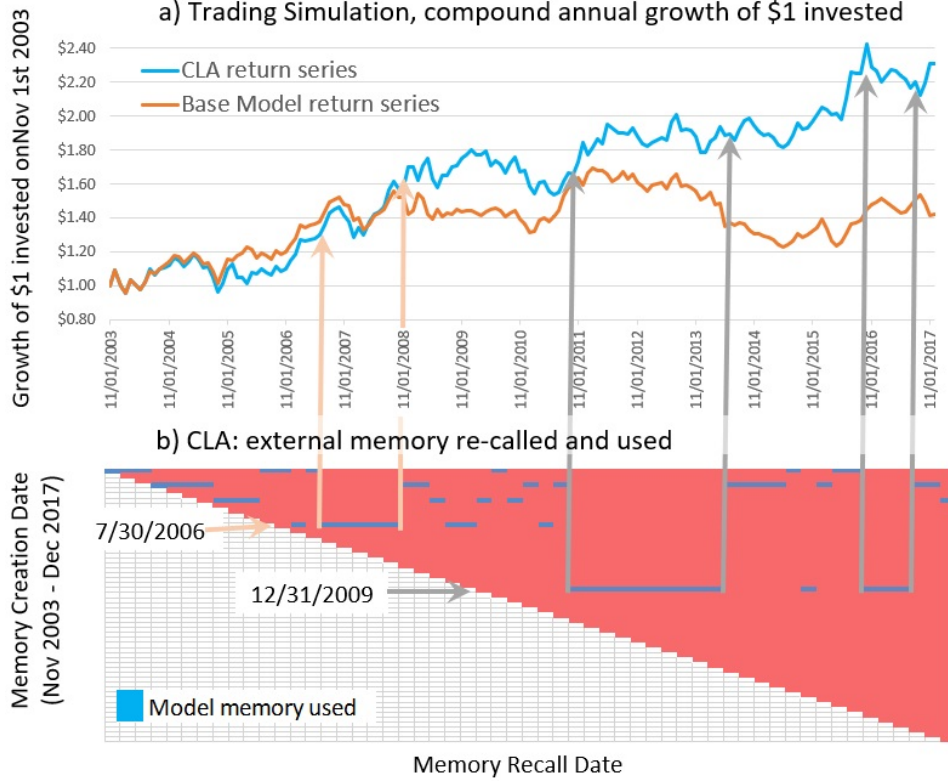
Figure 3: Interpretable memory: how recalled memories contribute to simulation performance. The graph in a) uses a single simulation and shows the growth of a $1 investment in 2003, using strategies driven by CLA or the base model. b) shows a representation of the CLA memory structure, where each row in the expanding triangle represents a potential memory. This external memory structure can grow by one memory at each step forward in the simulation, although in practice only four memories were remembered in this simulation. The top row in the triangular graphic represents the base model. The memory with the highest weight in each period is highlighted.

6 Conclusion

Continuous Learning Augmentation (CLA) is able to accumulate knowledge of changing market states over time and is able to apply this knowledge in approximately reoccurring future states. Our aim has been to combine different machine learning concepts to create a memory structure that improves the accuracy of time-series based modelling and produces memory modelling outcomes that are explainable. CLA is to our knowledge the first approach successfully applied to continuous learning in the noisy, non-stationary, finance problem space, using an explicit memory approach to drive state dependent decision making. We introduce absolute error change as a memory concept, using the hyperparameter J_{Crit} which learns points that govern remembering and forgetting of model-memories. We also use a sampling approach on multivariate DTW as a similarity measure to access CLA's memory structure and make the first application of a memory modelling approach to a broad stock selection problem.

6.1 Accuracy of outcomes

We find that CLA produces positive, statistically significant forecasting benefit using a FFNN base model. Long/short tests show positive and statistically significant total returns and a positive and statistically significant augmentation benefit relative to the base model. The similarity weighting of model-memories produces stronger results than simply picking the *best* model-memory. If CLA were exploited in practice, this outperformance would give significant advantage to investment strategy returns.

7

6.2 Explainability of memory use

CLA's memory structure can be interpreted in terms of which past state is relevant to forecasting in the current state. This allows objective comparisons to be made between relevant past states and the current state and also allows for a better understanding of the characteristics of the current state in the context of similar past states. This information can provide deep insights to users to guide decision making.

6.3 Future work

Our results indicate that CLA may be effectively applied to other problems on noisy and non-stationary time-series, in and outside of the finance domain. While our approach is directly applicable to quantitative investment we intend for our research to also be applied to other fields such as computational biology and wearable technology. It is also noted than the nature of absolute error change as a memory concept, introduced in this study, could hold more benefits for memory augmentation or model selection, where change in the absolute error *distribution* could be used to better identify changing states and to better learn more appropriate models. We feel the benefits of our approach that we established in terms of accuracy and explainability, justify the additional complexity of CLA and warrant research into a fully differentiable structure for learning the relationships between memory cues, memory models and ultimate actions.

References

[1] Evans J Brown R, Durbin J. Techniques for testing the constancy of regression relationships over time. *Journal of the Royal Statistical Society Series B (methodological)*, 37(2):149–192.

[2] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[3] Dan C. Ciresan, Ueli Meier, and Jurgen Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012.

[4] Picard D. Testing and estimating change-points in time series. *Advances in applied probability*.

[5] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. *Querying and mining of time series data: Experimental comparison of representations and distance measures*, volume 1, pages 1542–1552. 2 edition, 8 2008.

[6] Page E. On problems in which a change in a parameter occurs at an unknown point. *Biometrika*, 44(1/2):248–252.

[7] Smith A Engle R. Stochastic permanent breaks. *The Review of Economics and Statistics*, 81(4):553–574.

[8] F.J. Fabozzi, S.M. Focardi, and P.N. Kolm. *Quantitative Equity Investing: Techniques and Strategies*. Frank J. Fabozzi Series. Wiley, 2010.

[9] Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33:3–56, 1993.

[10] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. Technical report, 2017.

[11] French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 4:128–135, 1999.

[12] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA, 2006. ACM.

[13] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.

[14] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, October 2016.

[15] Yu H. High moment partial sum processes of residuals in arma models and their applications. *Journal of time series analysis*, 28(1):72–91.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[17] Schmidhuber J Hochreiter S. Long short term memory. *Neural Computation*, 9(8):1735–1780.

[18] Bai J. On the partial sums of residuals in autoregressive and moving average models. *Journal of Timeseries Analysis*, 14(3).

[19] V. K. Jandhyala and I. B. MacNeill. Residual partial sum limit process for regression models with applications to detecting parameter changes at unknown times. *Stochastic Processes and their Applications*, 33(2):309–323, 1989.

[20] MacNeill I Jandhyala V. The change point problem: a review of applications. *Developments in water science*, 27:381–387.

[21] M.W. Kadous, Mohammed Waleed Kadous, and Supervisor Claude Sammut. Temporal classification: Extending the classification paradigm to multivariate time series. Technical report, 2002.

[22] Lukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. *CoRR*, abs/1703.03129, 2017.

[23] Eamonn J. Keogh and Shruti Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Min. Knowl. Discov.*, 7(4):349–371, 2003.

[24] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.

[25] Zachary Chase Lipton, David C. Kale, Charles Elkan, and Randall C. Wetzel. Learning to diagnose with LSTM recurrent neural networks. *CoRR*, abs/1511.03677, 2015.

[26] Ian B. MacNeilt. Detecting unknown interventions with application to forecasting hydrological data. *JAWRA Journal of the American Water Resources Association*, 21(5):785–796, 1985.

[27] Seongsik Park, Sei Joon Kim, Seil Lee, Ho Bae, and Sungroh Yoon. Quantized memory-augmented neural networks. *CoRR*, abs/1711.03712, 2017.

[28] Jack W. Rae, Jonathan J. Hunt, Tim Harley, Ivo Danihelka, Andrew W. Senior, Greg Wayne, Alex Graves, and Timothy P. Lillicrap. Scaling memory-augmented neural networks with sparse reads and writes. *CoRR*, abs/1610.09027, 2016.

[29] Chib S. Estimation and comparison of multiple change point models. *Journal of Econometrics*, 86(2):221–241.

[30] H Sakoe and S Chiba.

[31] Skyler Seto, Wenyu Zhang, and Yichen Zhou. Multivariate time series classification using dynamic time warping template selection for human activity recognition. *CoRR*, abs/1512.06747, 2015.

[32] D Siegmund. Change-points: from sequential detection to biology and back. *Sequential Analysis*, 32(2-14):43–46.

[33] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017.

[34] Pablo Sprechmann, Siddhant M. Jayakumar, Jack W. Rae, Alexander Pritzel, Adrià Puig-domènech Badia, Benigno Uria, Oriol Vinyals, Demis Hassabis, Razvan Pascanu, and Charles Blundell. Memory-based parameter adaptation. *CoRR*, abs/1802.10542, 2018.

[35] Ogden R Sugiura, N. Testing change-points with linear trend. *Communications in Statistics B: Simulation and Computation*, 23:287–322.

[36] Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *CoRR*, abs/1606.04080, 2016.

[37] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014.

[38] Wojciech Zaremba and Ilya Sutskever. Reinforcement learning neural turing machines. *CoRR*, abs/1505.00521, 2015.

[39] Zheng Zhang, Romain Tavenard, Adeline Bailly, Xiaotong Tang, Ping Tang, and Thomas Corpetti. Dynamic time warping under limited warping path length. *Inf. Sci.*, 393(C):91–107, July 2017.

[40] Ji Hanlee Zhang N, Siegmund D and Li J. Detecting simultaneous change-points in multiple sequences. *Biometrika*, 97:631–646.